a compact topology on omega

Johan G. F. Belinfante
2004 October 12

```
In[1]:= SetDirectory["i:"]; << goedel62.10a; << tools.m

:Package Title: goedel62.10a          2004 October 10 at  9:25 p.m.

It is now:  2004 Oct 12 at 20:2

Loading Simplification Rules

TOOLS.M                      Revised 2004 September 25

weightlimit = 40
```

summary

This notebook contains a simple construction of a compact topology on the natural numbers, as well as some other related results. A class **x** is **compact** if every coarser collection is coarser than a finite collection. In this context, a **coarser collection** means a set **y** satisfying **subclass[y,x]** and **U[y] = U[x]**, that is,

```
In[2]:= member[pair[y, x], COARSER]

Out[2]= and[equal[U[x], U[y]], member[x, V], subclass[y, x]]
```

The definition of compact collection does not require that it be a topology. In the case that **x** is a topology, the underlying space is **U[x]**. In this case, any collection **y** coarser than **x** is conventionally called an **open cover** of the space, and a collection coarser than **y** is called a **subcover**. Note that in this context it is the class **U[x]** that is being covered, and not **x** itself.

omega

Every finite collection of sets is compact, so in particular every natural number is compact:

```
In[3]:=  SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u → omega, v → FINITE, w → COMPACT}]

Out[3]=  subclass[omega, COMPACT] == True

In[4]:=  subclass[omega, COMPACT] := True
```

---

omega is not compact

Since any collection is coarser than itself, any compact collection must be coarser than a finite one.

```
In[5]:=  subclass[COMPACT, image[COARSER, FINITE]]

Out[5]=  True
```

The class **omega** is not compact because there is no finite collection that is coarser than **omega**. To derive this fact, one starts by specializing a result about Uclosures of ordinal numbers to the special case of natural numbers:

```
In[6]:=  ImageComp[UCLOSURE, id[OMEGA], omega] // Reverse

Out[6]=  image[UCLOSURE, omega] == intersection[omega, complement[singleton[0]]]

In[7]:=  image[UCLOSURE, omega] := intersection[omega, complement[singleton[0]]]
```

As a corollary, one obtains:

```
In[8]:=  ImageComp[BIGCUP, inverse[S], omega] // Reverse

Out[8]=  image[BIGCUP, image[inverse[S], omega]] == omega

In[9]:=  image[BIGCUP, image[inverse[S], omega]] := omega
```

From this it follows that no collection coarser than **omega** can be finite.

```
In[10]:=  member[omega, image[COARSER, FINITE]] // AssertTest

Out[10]=  member[omega, image[COARSER, FINITE]] == False

In[11]:=  member[omega, image[COARSER, FINITE]] := False
```

The fact that **omega** is not compact follows:

```
In[12]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]],
           member[u, w], {u → omega, v → COMPACT, w → image[COARSER, FINITE]}]]

Out[12]= member[omega, COMPACT] == False

In[13]:= member[omega, COMPACT] := False
```

succ[omega]

The class **omega** is coarser than **succ[omega]**, but there is no finite set coarser than **omega**. Consequently, **succ[omega]** is not compact.

```
In[14]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → omega, v → image[inverse[COARSER], singleton[succ[omega]]],
            w → image[COARSER, FINITE]}]]

Out[14]= member[succ[omega], COMPACT] == False

In[15]:= member[succ[omega], COMPACT] := False
```

On the other hand, **succ[omega]** is a topology, and so this provides a simpe example of a topology that is not compact.

```
In[16]:= member[succ[omega], TOPS]

Out[16]= True
```

Since **singleton[omega]** is a finite collection that is coarser than **succ[omega]**, one has:

```
In[17]:= SubstTest[implies, and[member[pair[u, v], composite[Id, z]], member[u, x]],
          member[v, image[z, x]],
          {u → singleton[omega], v → succ[omega], x → FINITE, z → COARSER}]

Out[17]= member[succ[omega], image[COARSER, FINITE]] == True

In[18]:= member[succ[omega], image[COARSER, FINITE]] := True
```

The class **COMPACT** is sandwiched between **FINITE** and **image[COARSER, FINITE]**. Examples will be given to show that both of these inclusions are proper.

```
In[19]:= and[subclass[FINITE, COMPACT], subclass[COMPACT, image[COARSER, FINITE]]]

Out[19]= True
```

Since **succ[omega]** belongs to **image[COARSER,FINITE]** but not to **COMPACT**, it follows that the second inclusion is proper.

```
In[20]:= Map[not, SubstTest[implies, and[member[u, v], equal[v, w]], member[u, w],
           {u → succ[omega], v → image[COARSER, FINITE], w → COMPACT}]]

Out[20]= equal[COMPACT, image[COARSER, FINITE]] == False

In[21]:= equal[COMPACT, image[COARSER, FINITE]] := False
```

Membership in **COMPACT** means every coarser collection is coarser than a finite collection, whereas membership in **image[COARSER, FINITE]** is the weaker requirement that there is a coarser collection which is finite.

---

an infinite compact collection

The class **image[RC[omega], omega]** provides an example of an infinite compact collection. Each member of this class is the relative complement in **omega** of a natural number, that is, the set of all natural numbers greater than or equal to a given one. The proof that this collection is compact follows from the observation that any collection coarser than **image[RC[omega],omega]** must hold the set **omega** because that is the only set in the collection which holds the empty set. It follows that **singleton[omega]** is a finite collection coarser than any collection coarser than **image[-RC[omega],omega]**.

```
In[22]:= member[0, A[intersection[omega, x]]] // AssertTest

Out[22]= member[0, A[intersection[omega, x]]] == not[member[0, x]]

In[23]:= member[0, A[intersection[omega, x_]]] := not[member[0, x]]
```

Write **image[RC[omega],omega]** as the union of the complement of **0** and complements of nonzero numbers:

```
In[24]:= SubstTest[image, w, union[u, v],
           {u -> dif[omega, singleton[0]], v -> singleton[0], w → RC[omega]}] // Reverse

Out[24]= union[image[RC[omega], intersection[omega, complement[singleton[0]]]],
           singleton[omega]] == image[RC[omega], omega]

In[25]:= union[image[RC[omega], intersection[omega, complement[singleton[0]]]],
           singleton[omega]] := image[RC[omega], omega]
```

The latter complements do not hold **0**.

```
In[26]:= SubstTest[subclass, union[u, x], union[v, y],
          {u -> image[RC[omega], dif[omega, singleton[0]]],
           v -> P[complement[singleton[0]]],
           x → image[RC[omega], singleton[0]], y → singleton[omega]}]

Out[26]= subclass[image[RC[omega], omega],
           union[P[complement[singleton[0]]], singleton[omega]]] == True

In[27]:= % /. Equal → SetDelayed
```

This inclusion can be sharpened to an equation:

```
In[28]:= equal[intersection[complement[P[complement[singleton[0]]]],
             image[RC[omega], omega]], singleton[omega]] // AssertTest

Out[28]= equal[intersection[complement[P[complement[singleton[0]]]],
             image[RC[omega], omega]], singleton[omega]] == True

In[29]:= intersection[complement[P[complement[singleton[0]]]],
           image[RC[omega], omega]] := singleton[omega]
```

Introducing a variable **x**, one obtains:

```
In[30]:= Map[implies[#, equal[omega, x]] &, SubstTest[member, x, intersection[y, z],
             {y -> complement[P[complement[singleton[0]]]],
              z -> image[RC[omega], omega]}]] // Reverse

Out[30]= or[equal[omega, x], not[member[0, x]],
           not[member[intersection[omega, complement[x]], omega]],
           not[subclass[x, omega]]] == True

In[31]:= (% /. x → x_) /. Equal → SetDelayed
```

This says that **omega** is the only member of **image[RC[omega],omega]** that holds **0**.

```
In[32]:= implies[
          and[member[x, image[RC[omega], omega]], member[0, x]], equal[omega, x]]

Out[32]= True
```

From this will deduce that any collection **x** coarser than **image[RC[omega],omega]** must hold **omega.** The argument is this: if **U[x]** is omega, then **0** belongs to **U[x]**. So **0** must belong to some member of **x**. But that member of **x** must belong to **image[RC[omega],omega]**, and the only member of the latter that holds **0** is omega. So **omega** must belong to **x**.

```
In[33]:= SubstTest[implies, and[member[y, x], subclass[x, z]],
          member[y, z], z → image[RC[omega], omega]]

Out[33]= or[and[member[intersection[omega, complement[y]], omega], subclass[y, omega]],
           not[member[y, x]], not[subclass[x, image[RC[omega], omega]]]] == True

In[34]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed

In[35]:= Map[not, SubstTest[and, implies[and[p2, p3], p4],
           implies[and[p1, p4], p5], implies[and[p2, p5], p6],
           not[implies[and[p1, p2, p3], p6]], {p1 -> member[0, y],
            p2 → member[y, x], p3 → subclass[x, image[RC[omega], omega]],
            p4 → member[y, image[RC[omega], omega]],
            p5 → equal[y, omega], p6 → member[omega, x]}]]

Out[35]= or[member[omega, x], not[member[0, y]], not[member[y, x]],
           not[subclass[x, image[RC[omega], omega]]]] == True

In[36]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The variable **y** can be eliminated as follows:

```
In[37]:= Map[equal[V, #] &, SubstTest[class, y,
            or[member[omega, x], not[member[0, y]], not[member[y, x]],
             not[subclass[x, z]]]], z -> image[RC[omega], omega]]] // Reverse

Out[37]= or[member[omega, x], not[member[0, U[x]]],
           not[subclass[x, image[RC[omega], omega]]]] == True

In[38]:= (% /. x → x_) /. Equal → SetDelayed
```

If a covering of **omega** that holds **omega** has a finite subcovering, namely **singleton[omega]**.

```
In[39]:= SubstTest[implies, and[member[pair[u, v], composite[Id, z]], member[u, y]],
          member[v, image[z, y]], {u → singleton[omega], v → x, y → FINITE, z → COARSER}]

Out[39]= or[member[x, image[COARSER, FINITE]], not[equal[omega, U[x]]],
           not[member[omega, x]], not[member[x, V]]] == True

In[40]:= (% /. x → x_) /. Equal → SetDelayed
```

These results can be combined:

```
In[41]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3],
           implies[and[p1, p3], p4], implies[p4, p5], not[implies[p1, p5]], {p1 ->
             member[x, image[inverse[COARSER], singleton[image[RC[omega], omega]]]],
            p2 → member[0, U[x]], p3 → member[omega, x],
            p4 → member[pair[singleton[omega], x], COARSER],
            p5 → member[x, image[COARSER, FINITE]]}]]

Out[41]= or[member[x, image[COARSER, FINITE]], not[equal[omega, U[x]]],
           not[member[x, V]], not[subclass[x, image[RC[omega], omega]]]] == True

In[42]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. The set **image[RC[omega], omega]** is a compact collection.

```
In[43]:= Map[equal[V, #] &, SubstTest[class, x,
            implies[member[x, u], member[x, image[COARSER, FINITE]]],
            u -> image[inverse[COARSER], singleton[image[RC[omega], omega]]]]] //
          Reverse

Out[43]= member[image[RC[omega], omega], COMPACT] == True

In[44]:= member[image[RC[omega], omega], COMPACT] := True
```

This compact set is not finite.

```
In[45]:= Map[not, SubstTest[implies, and[FUNCTION[x], member[y, FINITE]],
           member[image[x, y], FINITE], {x -> RC[omega], y → image[RC[omega], omega]}]]

Out[45]= member[image[RC[omega], omega], FINITE] == False

In[46]:= member[image[RC[omega], omega], FINITE] := False
```

It follows that the inclusion **subclass[FINITE, COMPACT]** is a proper one.

```
In[47]:= Map[not, SubstTest[implies, and[member[u, v], equal[v, w]],
           member[u, w], {u → image[RC[omega], omega], v → COMPACT, w → FINITE}]]

Out[47]= equal[COMPACT, FINITE] == False

In[48]:= equal[COMPACT, FINITE] := False
```

---

a basis for an infinite compact topology

The class **image[RC[omega], omega]** is totally ordered by inclusion. To derive this fact, one begins with the fact that **omega** is totally ordered by inclusion:

```
In[49]:= SubstTest[implies, subclass[u, v],
          subclass[image[w, u], image[w, v]], {u → cart[omega, omega],
           v → union[S, inverse[S]], w → cross[RC[omega], RC[omega]]}]

Out[49]= subclass[cart[image[RC[omega], omega], image[RC[omega], omega]], union[
            composite[id[P[omega]], S], composite[inverse[S], id[P[omega]]]]] == True

In[50]:= % /. Equal → SetDelayed
```

This result just needs to be cleaned up:

```
In[51]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
          {u -> cart[image[RC[omega], omega], image[RC[omega], omega]],
           v -> union[composite[id[P[omega]], S], composite[inverse[S], id[P[omega]]]],
           w → union[S, inverse[S]]}]

Out[51]= subclass[cart[image[RC[omega], omega], image[RC[omega], omega]],
           union[S, inverse[S]]] == True

In[52]:= subclass[cart[image[RC[omega], omega], image[RC[omega], omega]],
           union[S, inverse[S]]] := True
```

It follows that this class is closed under binary intersections, and is therefore a basis for a topology.

```
In[53]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
          {u -> image[RC[omega], omega],
           v -> cliques[union[S, inverse[S]]], w → binclosed[CAP]}]

Out[53]= subclass[image[CAP, cart[image[RC[omega], omega], image[RC[omega], omega]]],
           image[RC[omega], omega]] == True

In[54]:= subclass[image[CAP, cart[image[RC[omega], omega], image[RC[omega], omega]]],
           image[RC[omega], omega]] := True
```

This class is also closed under binary unions:

```
In[55]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
          {u -> image[RC[omega], omega],
           v -> cliques[union[S, inverse[S]]], w → binclosed[CUP]}]

Out[55]= subclass[image[CUP, cart[image[RC[omega], omega], image[RC[omega], omega]]],
           image[RC[omega], omega]] == True

In[56]:= subclass[image[CUP, cart[image[RC[omega], omega], image[RC[omega], omega]]],
           image[RC[omega], omega]] := True
```

Although the class **image[RC[omega],omega]** is closed under binary unions, it is not closed under arbitrary unions for the simple fact that it fails to hold the union of the empty set. To obtain a topology, one therefore needs to add the empty set.

---

topology

One obtains a topology from a topological base by applying **Uclosure**:

```
In[57]:= Map[implies[member[x, y], #] &,
          SubstTest[implies, and[member[x, u], subclass[u, v]], member[x, v],
           {u → binclosed[CAP], v → image[inverse[UCLOSURE], TOPS]}]]

Out[57]= or[member[Uclosure[x], TOPS], not[member[x, y]],
           not[subclass[image[CAP, cart[x, x]], x]]] == True

In[58]:= or[member[Uclosure[x_], TOPS], not[member[x_, y_]],
           not[subclass[image[CAP, cart[x_, x_]], x_]]] := True
```

To carry out this idea, one needs a formula for the **Uclosure** of the class **image[-RC[omega], omega]**. The net result, as we shall see momentarily, amounts to adding the empty set.

```
In[59]:= SubstTest[image, RC[omega], union[omega, x], x → singleton[omega]]

Out[59]= image[RC[omega], succ[omega]] == union[image[RC[omega], omega], singleton[0]]

In[60]:= image[RC[omega], succ[omega]] := union[image[RC[omega], omega], singleton[0]]
```

Some rewrite rules will be needed to simplify various expressions that appear in the derivation.

```
In[61]:= equal[intersection[P[omega], succ[omega]], succ[omega]]

Out[61]= True
```

This fact is made into a rewrite rule:

```
In[62]:= intersection[P[omega], succ[omega]] := succ[omega]
```

Here is another such fact, which is also made into a rewrite rule.

```
In[63]:= equal[intersection[P[omega], Uclosure[image[RC[omega], x]]],
          Uclosure[image[RC[omega], x]]]

Out[63]= True

In[64]:= intersection[P[omega], Uclosure[image[RC[omega], x_]]] :=
          Uclosure[image[RC[omega], x]]
```

For finite sets, there is no difference between **Aclosure[x]** and **fix[HULL[x]]**. In particular:

```
In[65]:= SubstTest[implies, member[x, V],
          equal[fix[HULL[x]], Aclosure[x]], x → succ[omega]]

Out[65]= equal[fix[HULL[succ[omega]]], succ[omega]] == True

In[66]:= fix[HULL[succ[omega]]] := succ[omega]
```

The needed **Uclosure** formula now follows:

```
In[67]:= Map[fix, Map[composite[RC[omega], #, RC[omega]] &, SubstTest[HULL,
              image[RC[omega], x], x → image[RC[omega], succ[omega]]]]] // Reverse

Out[67]= Uclosure[image[RC[omega], omega]] ==
          union[image[RC[omega], omega], singleton[0]]

In[68]:= Uclosure[image[RC[omega], omega]] :=
          union[image[RC[omega], omega], singleton[0]]
```

This yields the promised example of a compact topology on the natural numbers:

```
In[69]:= SubstTest[implies, member[x, binclosed[CAP]],
          member[Uclosure[x], TOPS], x → image[RC[omega], omega]]

Out[69]= member[union[image[RC[omega], omega], singleton[0]], TOPS] == True

In[70]:= member[union[image[RC[omega], omega], singleton[0]], TOPS] := True
```

This is indeed a topology on the set **omega**:

```
In[71]:= U[union[image[RC[omega], omega], singleton[0]]]

Out[71]= omega
```

The statement that this topology is compact is recognized automatically from a rewrite rule that says that compactness is not affected by adding the empty set.

```
In[72]:= member[union[image[RC[omega], omega], singleton[0]], COMPACT]

Out[72]= True
```

This topology, is of course, not finite:

```
In[73]:= member[union[image[RC[omega], omega], singleton[0]], FINITE]

Out[73]= False
```

Comment. The topology that has been constructed in this notebook bears some resemblance to the cofinite topology, **Uclosure[image[RC[omega], FINITE]]**. The latter is also compact, but the derivation of that fact is not as simple.